UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/004,196	11/14/2001	Jose Fernandez	42390P10728	5794

8791 7590 02/26/2008
BLAKELY SOKOLOFF TAYLOR & ZAFMAN
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040

EXAMINER	
GUILL, RUSSELL L	

ART UNIT	PAPER NUMBER
2123	

MAIL DATE	DELIVERY MODE
02/26/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

mn

Office Action Summary	Application No.	Applicant(s)	
	10/004,196	FERNANDEZ, JOSE	
	Examiner	Art Unit	
	RUSSELL GUILL	2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *18 December 2007*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-5 and 7-49* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *1-5 and 7-42* is/are allowed.

6)☒ Claim(s) *43-49* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *20 December 2005* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) □ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) □ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) □ Notice of Informal Patent Application
6) □ Other: _____.

DETAILED ACTION

1. This action is in response to an **Appeal Brief** filed December 18, 2007.

2. All appealed claims appear to be allowable over the prior art of record, specifically claims 29 - 35 rejected under 35 U.S.C. § 101, and claims 1 - 3, 7 - 13, 15 - 20 and 22 - 23, and their dependent claims 4 - 5, 14, 21, and 24 - 28, and claims 29 - 42 which were rejected under 35 U.S.C. § 103. Claims 43 - 49, which were not appealed, remain rejected, as described below.

3. In view of the appeal brief filed on December 18, 2007, PROSECUTION IS HEREBY REOPENED. To avoid abandonment of the application, appellant must exercise one of the following two options: (1) file a reply under 37 CFR 1.111 (if this Office action is non-final) or a reply under 37 CFR 1.113 (if this Office action is final); or, (2) initiate a new appeal by filing a notice of appeal under 37 CFR 41.31 followed by an appeal brief under 37 CFR 41.37. The previously paid notice of appeal fee and appeal brief fee can be applied to the new appeal. If, however, the appeal fees set forth in 37 CFR 41.20 have been increased since they were previously paid, then appellant must pay the difference between the increased fees and the amount previously paid. A Supervisory Patent Examiner (SPE) has approved of reopening prosecution by signing below:

PAUL RODRIGUEZ
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Response to Arguments

4. Regarding **claims 29 - 35** rejected under 35 U.S.C. § 101:

4.1. Applicant's arguments have been fully considered, and are persuasive.

5. Regarding independent claim 1 rejected under 35 U.S.C. § 103:

5.1. Applicant's arguments have been fully considered, and claim 1 appears to be allowable over the prior art of record, as follows. Parts of the argument were not persuasive, as follows.

5.2. The Applicant argues:

5.3. Manning shows an indexing system for aiding in queries. As stated in the abstract, one entry is added to at least one table. These tables, shown in Figure 6, are used in executing queries.

The tables index the contents of XML documents (paragraph 6, line3, see also paragraph 9). and facilitate flexible queries to extract data from real and virtual documents (paragraph 7). There is no mention of persistence packages, persistent data, nor transforms being applied to format data. Instead, Manning adds tables with pointers to the undisturbed XML data to be used for queries in an RDBMS.

5.3.1. The Examiner respectfully replies:

5.3.2. Applicant's argument is not persuasive. The specification recites, "Persistence is a property related to programming languages by which created objects continue to exist and variables continue to retain their values between runs of a program."

5.3.3. Please refer to Manning, paragraphs [0004] and [0028]. An XML document is received and stored in a relational database. The XML document is a data object, and contains a Document Type Definition (DTD) that provides attributes for each element in the document, and indicates the relationship of the elements; that is, the DTD is metadata. The DTD is included with the received document. Thus, the XML document is a persistence package with persistent data. Further, the invention of Manning refers to structured documents in general, not just XML documents (please see paragraph [0008], "other interchangeable structured document formats").

5.3.4. Further, Manning appears to use transforms. The specification recites that a transform establishes a storage format and/or storage location for the persistent data. Please refer to paragraph [0028]; paragraph [0041]; figure 3, elements 102 - 110; and paragraph [0009], "At least one table is generated based on a schema of elements . . . ". First, Manning clearly uses a database schema to determine a storage location and format for the stored data, thus the schema is a transform. It was common knowledge in the art at the time of invention that a database schema determined a storage location and format for the stored data; please refer to the new art:

5.3.4.1. Sams, "Sams Teach Yourself PL/SQL™ in 21 Days", second edition, 2000, Sams Publishing; teaches a database schema establishing a storage location (table) and data storage formats for the data in the table (page 224, listing 8.1 Creating the Employee Table), and teaches automatic data format conversion by a database system (*page 224 - 225, section "Using the INSERT Statement", please note that the pay rate being inserted into the employee table is a text ('8.50') but the pay rate is inserted as a number in the table, where* *the pay_rate is defined as a number in the schema on page 224, listing 8.1 Creating the Employee Table).*

5.3.5. Further, transforms were common knowledge in the art at the time of invention. Please refer to the new art:

5.3.5.1. SqlDatabaseLanguage ("Database Language SQL - Part 2: Foundation (SQL/Foundation)", September 1999, ISO/IEC 9075-2:1999) teaches transforms that determine a storage format for data.

5.4. The Applicant argues:

5.5. In Claim 1, as amended, for example, persistent data is extracted from the persistence package. This data is then formatted and stored accordingly. In Manning, there is no formatting of the data before it is stored. Instead, some of the metadata is copied into tables. Accordingly, Claim 1 is believed to be allowable over the reference.

5.5.1. The Examiner respectfully replies:

5.5.2. Applicant's argument is not persuasive. Please refer to the Examiner's reply above regarding persistent data.

5.5.3. The specification does not appear to define the term "format", and so a plain language meaning is applied. Manning clearly formats and stores data. First, refer to the Examiner's reply above. Further, at the very least, Manning changes the format of the data from an XML+DTD format into a relational database format (figure 5 versus figure 7, and paragraph [0004]).

5.6. The Applicant argues:

5.7. In paragraph 3.1, the Examiner sets forth an interpretation of Manning as showing a persistence package, persistent data, and a storage format. This interpretation does not come from the reference. However, due to the Examiner's persistence in this argument, Applicant will discuss the reference as if the claims applied to Manning.

5.7.1. The Examiner respectfully replies:

5.7.2. Applicant's argument is not persuasive. This argument does not appear to need a reply, however, please refer to the Examiner's replies above regarding a persistence package, persistent data and storage format.

5.8. The Applicant argues:

5.9. In paragraph 3.2, the Examiner continues to draw the analogy in terminology between the claims and Manning as applied to "transforms". The Examiner further suggests that the values in the page table are formatted or transformed because the page table 220 values are numbers in the NUMBER column rather than text. Applicant submits that this is a table of page numbers. The values in the original document would be page numbers, just as they are page numbers in the NUMBER column. This may be compared to the text object table in which the data values are indicated as being text. The page table 220 is further illuminated by reference to Figure 5 in which there is an item 1006, identified as page number and the value is "2". "2" is the number from the document that will be added to the number field in the page table.

5.10. In paragraph 3.3, the Examiner continues that the NUMBER column shows that a value has been transformed or formatted as a number rather than as text. Again, Applicant submits that the value 1006 in the original is also a number not text.

5.10.1. The Examiner respectfully replies:

5.10.2. Applicant's argument is not persuasive. Please refer to the Examiner's reply above regarding transforms.

5.10.3. Further, as discussed above, the specification does not appear to define the term "format", and so a plain language meaning is applied. Manning clearly formats and stores data. First, refer to the Examiner's reply above. Further, at the very least, Manning changes the format of the data from an XML+DTD format into a relational database format (figure 5 versus figure 7, and paragraph [0004]).

5.10.4. As recited by the Applicant above, "The page table 220 is further illuminated by reference to Figure 5 in which there is an item 1006, identified as page number and the value is "2".". Thus the Applicant admits that the value "2" is text data. The value is stored in page table 220 as a NUMBER – please compare with table 218 where text data is stored in quotations. Further, the ordinary artisan would have known that text numbers are converted to number format, as shown in the following art that teaches common knowledge:

5.10.4.1. Sams, "Sams Teach Yourself PL/SQL™ in 21 Days", second edition, 2000, Sams Publishing; teaches a database schema establishing a storage location (table) and data storage formats for the data in the table (page 224, listing 8.1 Creating the Employee Table), and teaches automatic data format conversion by a database system (*page 224 – 225, section "Using the INSERT Statement", please note that the pay rate being inserted into the employee table is a text ('8.50') but the pay rate is inserted as a number in the table, where the pay_rate is defined as a number in the schema on page 224, listing 8.1 Creating the Employee Table*).

5.11. The Applicant argues:

5.12. By contrast, in Claim 1, the persistence package is in a format of a software component and is transformed into a storage format that is independent of the software component that provided the data. In Manning, the data is in XML, an application-independent language that is designed to be used by a wide range of different applications. It is not formatted for any particular software component. It is then converted into SQL tables, another application-independent structure.

5.12.1.1. The Examiner respectfully replies:

5.12.1.2. Applicant's argument is not persuasive. Please refer to the XML data object of figure 5. The data object was generated by a software component, and is thus in the format of the software component, which happens to be XML. Further, the Applicant appears to be using the word "format" with multiple meanings. Previously the Applicant argued that data in a table was not in a number format, but now the word "format" appears to have changed meaning, and appears to apply to the structure of a record.

5.13. The Applicant argues:

5.14. In Claim 1, the data is provided from one of a plurality of different software components having persistent data in different formats. In Manning, the data is all in the same XML format no matter where it came from.

5.14.1. The Examiner respectfully replies:

5.14.2. Applicant's argument is not persuasive. Manning clearly teaches different formats for the data objects. First, Manning teaches data such as vector graphics, e-commerce transactions, mathematical equations, object meta-data (*paragraph [0004]*), which are clearly different formats. Further, Manning teaches different formats for data objects in paragraph [0008], "and other interchangeable structured document formats". The invention of Manning is clearly directed to these other interchangeable structured document formats. Further, Manning appears to teach that the XML documents (including data objects encoded as XML) are available over the Internet at paragraph [0006], first sentence.

5.15. The Applicant argues:

5.16. In Claim 1, the data is transformed without using the software component from which the persistence package is received. In Manning, different software components are not identified, but it is probable that the entire system is designed to work with the same XML formatted documents using the same software components.

5.16.1. The Examiner replies:

5.16.2. Applicant's argument is not persuasive. Manning teaches different formats for data objects in paragraph [0008], "and other interchangeable structured document formats". The invention of Manning is clearly directed to these other interchangeable structured document formats. First, Manning teaches data such as vector graphics, e-commerce transactions, mathematical equations, object meta-data (*paragraph [0004]*), which are clearly different formats.

5.17. The Applicant argues:

5.18. Also in Claim 1, the storage format is a format that is compatible with the receiving system and with a storage device independent of the software component from which the persistent package was received. Manning does not say whether any software components are to be used but many SQL systems also handle XML.

5.18.1. The Examiner respectfully replies:

5.18.2. Applicant's argument is not persuasive. Manning teaches different formats for data objects in paragraph [0008], "and other interchangeable structured document formats". The invention of Manning is clearly directed to these other interchangeable structured document formats.

5.18.3. Further, the Applicant alleges that SQL systems handle XML, but provides no support for the allegation at the time of invention. Further, it is clear from Manning, that the invention is not using any automatic XML features.

5.18.4. Further, please refer to the recited portion of the claim below:

5.18.4.1. format of the software component into a storage format that is compatible with the receiving system and with a storage device of the running system independent of the software component (***paragraph [0022], paragraphs [0027] - [0029], and paragraph [0034]; the received data such as vector graphics and e-commerce transactions (see paragraph [0004]) is being stored in a relational database, which is a storage format that is compatible with the receiving system and with a storage device independent of the software component. Also, see figure 5, displaying a received data format, and figure 7, the corresponding stored data format. The data is stored in tables, which is a different format than the received format. Further, element 1003 is a text data, which is stored in page table 220 as a number. Please compare with table 218 where text data is stored in quotations. Further, the ordinary artisan would have known that text numbers are converted to number format.***)

5.19. The Applicant argues:

5.20. As has been previously explained, Manning does not show formatting data before it is stored. Instead, in Manning some of the metadata is copied into tables. The Examiner's response to this fact of the reference is that "it would have been obvious." The Examiner's argument now asserted through more than 100 pages of written Office actions is to take words in Manning out of context and fill in with "it would have been obvious."

5.20.1. Applicant's argument is not persuasive. Formatting of the data has been discussed in previous Examiner's replies.

5.20.2. Regarding, "in Manning some of the metadata is copied into tables". Some of the metadata is copied into tables, but also, the data itself is copied into tables, as shown between figure 5 and figure 7.

5.20.3. Regarding "it would have been obvious", please refer to the following principles:

5.20.3.1. A claimed invention is not patentable if the subject matter of the claimed invention would have been obvious to a person having ordinary skill in the art. *KSR International Co. v. Teleflex Inc.*, 127 S. Ct. 1727 (2007).

5.20.3.2. The question under 35 U.S.C. § 103 is not merely what the references teach but what they would have suggested to one of ordinary skill in the art at the time the invention was made. *In re Lamberti*, 545 F.2d 747, 750 (CCPA 1976).

5.20.3.3. One of ordinary skill in the art is presumed to have skills apart from what the prior art references expressly disclose. *In re Sovish*, 769 F.2d 738, 742-743 (Fed. Cir. 1985).

5.20.3.4. A person of ordinary skill is also a person of ordinary creativity, not an automaton. *KSR*, 127 S. Ct. at 1742.

5.20.3.5. The combination of familiar elements according to known methods is likely obvious when the combination does no more than yield predictable results. *KSR*, 127 S. Ct. at 1739.

5.21. The Applicant argues:

5.22. The Examiner's "it would have been obvious" approach is discussed in more detail below:

5.22.1. The Examiner respectfully replies:

5.22.2. No reply appears needed.

5.23. The Applicant argues:

5.24. Claim 1 of the present application begins with receiving a persistence package from one of a plurality of different software components, the software components having persistent data in different formats. First, the Examiner has ignored the words "persistence" and "persistent" which have no parallel in Manning. Second, the Examiner has inferred the software components.

5.24.1. The Examiner respectfully replies:

5.24.2. Applicant's argument is not persuasive. As recited above, the specification recites, "Persistence is a property related to programming languages by which created objects continue to exist and variables continue to retain their values between runs of a program."

5.24.3. Please refer to Manning, paragraphs [0004] and [0028]. An XML document is received and stored in a relational database. The XML document is a data object, and contains a Document Type Definition (DTD) that provides attributes for each element in the document, and indicates the relationship of the elements; that is, the DTD is metadata. The DTD is included with the received document. Thus, the XML document is a persistence package with persistent data. Further, the invention of Manning refers to structured documents in general, not just XML documents (please see paragraph [0008], "other interchangeable structured document formats").

5.24.4. Further, it clearly appears reasonable to infer software components from the art of Manning. At the least, an ordinary artisan would have known that software components are used to create data objects, as discussed in paragraph [0004].

5.25. The Applicant argues:

5.26. Most importantly, the Examiner asserts that ""it would have been obvious that persistent data from vector graphics is in a different format than e-commerce transactions." Applicant shall assume that the Examiner means to take Official Notice that data from vector graphics is in a different format than data from e-commerce transactions. Be that as it may, in Manning all the data is in XML + DTD format. There is nothing in the reference to suggest that there are different software components with different data formats.

5.26.1. The Examiner respectfully replies:

5.26.2. Applicant's argument is not persuasive. As recited previously, Manning teaches different formats for data objects in paragraph [0008], "and other interchangeable structured document formats", and the previously recited vector graphics and e-commerce transactions. The invention of Manning is clearly directed to these other interchangeable structured document formats.

5.27. The Applicant argues:

5.28. Moving on to Claim 1's recitation of "establishing, based on the extracted metadata, a transform for a storage format for the persistent data," the Examiner leaves out the word "transform" and points to paragraphs 28 and 41. These paragraphs refer only to creating tables. The tables contain information taken from the XML data. The tables allow for querying and do not appear to have anything to do with persistence or different software applications, nor with transforms. The tables would appear to be additional metadata (para. 28, lines 3-4). At this point, Applicant is uncertain what "transform," and "storage format" are being read on. "Based on the extracted metadata" seems to have been ignored.

5.28.1. The Examiner respectfully replies:

5.28.2. Applicant's argument is not persuasive. First, the specification recites, "Transforms establish a storage format and/or storage location for the persistent data". Therefore, the recited paragraphs 41 and 28 are used to show that the metadata is used to establish a storage format and/or storage location for the persistent data. In summary, in Manning, each record has a Document Type Description (DTD) that is metadata. The DTD is used in paragraph 28 to generate a table for each element in the DTD, where each element table includes a column for each object, e.g. attribute or content, defined for the element. This defines a storage format and/or storage location for the persistent data, and is therefore a transform. Also, it is implied that a database schema is created in this process, which establishes a storage format. Also, as recited in the Office Action, please refer to figure 3, elements 114 - 128.

5.28.3. Further, transforms were common knowledge in the art at the time of invention. Please refer to the new art:

5.28.3.1. SqlDatabaseLanguage ("Database Language SQL - Part 2: Foundation (SQL/Foundation)", September 1999, ISO/IEC 9075-2:1999) teaches transforms that determine a storage format for data.

5.29. The Applicant argues:

5.30. Claim 1 next recites, "applying the transform to the persistent data to format the persistent data." Manning does no such thing. The data is untouched, it is complemented by the element directory table and the navigation table, there is no transform to apply and no formatting.

5.30.1. The Examiner respectfully replies:

5.30.2. Applicant's argument is not persuasive. At the least, in figure 3, element 124, and paragraph 29, the data is extracted from the XML record and stored in the database. This process certainly applies the storage format (i.e., transform) to the persistent data to format the persistent data without using the software component from which the persistence package was received during the runtime of the receiving system from the format of the software component (according to Applicant above, XML+DTD format) into a storage format that is compatible with the receiving system and with a storage device independent of the software component. At the least, the data from the original record is no longer in XML+DTD format, and therefore is formatted for the storage in the receiving system. In addition to the data stored, there is also an element directory table and the navigation table, as described in paragraph [0032].

5.31. The Applicant argues:

5.32. Claim 1 further defines this formatting to be "from the format of the software component into a storage format." As mentioned previously, the Manning data comes in as XML and stays as XML with a little more metadata.

5.32.1. The Examiner respectfully replies:

5.32.2. Applicant's argument is not persuasive. As discussed above, the data appears to be changed from XML to an internal format of the receiving system, as shown at least by figure 7, compared to the XML in figure 5. Further, formatting has been thoroughly discussed in the Examiner's replies above.

5.33. The Applicant argues:

5.34. The Examiner has responded that "it would have been obvious that a transform is applied" Applicant believes that the Examiner means to assert that it is inherent in Manning that when a value is stored in an element table, a transform is applied to the value to determine where in the table to store that value. Of course, there is no suggestion in Manning that a transform be used. It would appear that the tables are added to the XML and that the database uses the XML tables.

5.34.1. The Examiner respectfully replies:

5.34.2. Applicant's argument is not persuasive. The specification appears to recite, "Transforms establish a storage format and/or storage location for the persistent data". At the least, the data from the original record is no longer in XML+DTD format, and therefore is formatted for the storage in the receiving system, as shown at least by figure 7, compared to the XML in figure 5. Therefore, a transform was applied to change the format, even though Manning does not explicitly mention a transform.

5.34.3. Further, transforms were common knowledge in the art at the time of invention. Please refer to the new art:

5.34.3.1. SqlDatabaseLanguage ("Database Language SQL – Part 2: Foundation (SQL/Foundation)", September 1999, ISO/IEC 9075-2:1999) teaches transforms that determine a storage format for data.

5.34.4. Further, as discussed previously, Manning clearly uses a database schema to determine a storage location and format for the stored data, thus the schema is a transform. As discussed previously, it was common knowledge in the art at the time of invention that a database schema determined a storage location and format for the stored data.

5.34.5. Regarding the Applicant's statement, "It would appear that the tables are added to the XML and that the database uses the XML tables.", the Examiner cannot find any support for the statement in Manning. Further, the Examiner does not understand how to add tables to XML. The statement is not supported, and is thus simply an allegation.

5.35. The Applicant argues:

5.36. As to the storing element of Claim 1, the Examiner would appear to be asserting that it is inherent in Manning that the tables are stored, that the storing is done in a storage device and that this is done during runtime. There are limitations regarding persistence, software components, and the storage device that all interrelate and that are being ignored.

5.36.1. The Examiner respectfully replies:

5.36.2. Applicant's argument is not persuasive. The storing element of claim 1 appears to be, "storing the persistent data in the storage device in the storage format". Paragraph [0029] of Manning clearly recites storing the persistent data, "inserts in the added entry . . . each object (e.g., attribute value or content) for the element into the corresponding object column of the element table". Since the data is being stored in a relational database, this would reasonably suggest the limitation to the ordinary artisan.

5.37. The Applicant argues:

5.38. For further support regarding the transforms, the Examiner has tumed to Morgenstem at Col. 6, line 1, through Col. 8, line 53. Applicant is unable to find a transform for a storage format for persistent data in this section of Morgenstem. Applicant is further unable to find, and the Examiner is unable to point out, any suggestion that transformation generator 42 establishes a transform based on metadata or operates without using the software components from which a data package is received.

5.39. On the contrary, it would appear that the source is tightly connected with the transform generator and transformer engine 66.

5.39.1. The Examiner respectfully replies:

5.39.2. Applicant's argument is not persuasive. First, the Office Action appears to reference (in Morgenstern) Col. 8, lines 53 - 67, and Col. 9, lines 1 - 3, and Col. 7, lines 16 - 67, and Col. 8, lines 1 - 53, rather than Col. 6, line 1, through Col. 8, line 53, as recited above. Further, the Office Action references elements in figure 2. Significant reference material is included in the missing portions of the references. However, the Office Action uses Morgenstern to provide a transform. The specification recites, "Transforms establish a storage format and/or storage location for the persistent data". In figure 2, element 22, a hi-level data schema specification is used as input to a transform generation module 30. The hi-level data schema specification 22 is metadata. The output of the generation module 30 is an information bridge mediator 60, which transforms source data 62 into target data 70. The target data is in a different format than the source data, and therefore, the information bridge mediator 60, which transforms source data 62 into target data 70, is a transform.

5.40. The Applicant argues:

5.41. Further, limitations in the claims additionally distinguish the invention from the references. Here are a few:

5.42. "foreign to the running system

5.43. "transforms establish a storage format and/or "storage location"

5.44. "description of the format of the persistent data"

5.45. Applicant respectfully submits that these additional features reflect features not present in the cited combination and accordingly, the rejection is respectfully traversed. For example, the XML documents require a specific program to parse the XML. This program is normally also able to edit, display, print, and convert the documents and style sheets, etc. The software is thus not foreign to the system.

5.45.1. The Examiner respectfully replies:

5.45.2. Since the Applicant does not provide any support for the statements, "transforms establish a storage format and/or "storage location" and "description of the format of the persistent data", they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

5.45.3. Regarding the Applicant's statements, "the XML documents require a specific program to parse the XML. This program is normally also able to edit, display, print, and convert the documents and style sheets, etc. The software is thus not foreign to the system.". The Applicant provides no support for the allegation that the program is normally able to edit, display, print, and convert the documents and style sheets, and further, the Examiner respectfully disagrees with the allegation. However, upon further consideration, it appears that it may not be obvious that the software is not foreign to the system, and thus, the claim is allowable over the prior art of record.

5.46. The Applicant argues:

5.47. Manning and Morgenstern are both directed to indexing systems for queries in a relational database. The invention of Claim 1 is directed, at least in part, to formatting data from the format of the software component from which it was received into a storage format that is compatible with the receiving system. This is done without using the software component.

5.47.1. The Examiner respectfully replies:

5.48. Applicant's argument is not persuasive. As discussed above, Manning formats data from the format of the software component from which it was received into a storage format that is compatible with the receiving system without using the software component.

5.49. The Applicant argues:

5.50. Accordingly, the elements of Claim 1 are not met by the cited combination. Claims 29 and 36 were rejected on similar grounds. These claims contain even more limitations that are neither taught nor suggested by the combination.

5.50.1. The Examiner respectfully replies:

5.50.2. As recited above, claim 1 appears that it may be allowable over the prior art of record.

5.50.3. Claims 29 and 36 do not appear to have been under appeal.

5.50.4. The remaining limitations are not argued, and thus are not relevant.

Claim Objections

6. Claim 9 is objected to for the following minor informality: the claim recites, "a description of the format of the persistent data the software components having persistent data". The phrase appears to want a comma. The phrase appears to mean, "a description of the format of the persistent data**,** the software components having persistent data ".

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. **Claims 43 – 44 and 47 - 49** are rejected under 35 U.S.C. 103(a) as being unpatentable over Manning (U.S. Patent Application Publication Number US 2002/0103829) in view of Maimone (U.S. Patent Number 6,418,451).

8.1. The art of Manning is directed to managing structured documents in a database (*abstract*), the structured documents in a variety of structured formats (*paragraph [0008], "and other interchangeable structured document formats"; and paragraph [0012], "In further implementations, the structured document comprises an Extensible Markup Language (XML) document", XML is only one example of a structured document*) including persisting data objects in a database (*paragraph [0034], an XML document is a data object*).

8.2. The art of Maimone is directed to persisting objects in a relational database (*title and abstract*).

8.3. The art of Manning and the art of Maimone are analogous art at least because they both pertain to the art of persisting data objects in a database.

8.4. Regarding claim 43, Manning appears to teach:

8.4.1. a machine-readable medium comprising instructions that are executed by a machine *(paragraphs [0021] and [0022])*.

8.4.1.1. Regarding *(paragraphs [0021] and [0022])*; it would have been obvious that instructions cause a machine to execute because a computer was a machine, and computers execute instructions.

8.4.2. receiving persistent data having a model structure *(figure 3, item 100 - 102; please note that a DTD is a Document Type Definition that provides attributes for each element in the document, and indicates the relationship of the elements – please refer to paragraph [0004]; and paragraph [0028]; please note that the DTD is included with the received document)* from one of a plurality of different software components (*paragraph [0004] first and second sentences; it would have been obvious that the multiple data types recited, such as vector graphics and e-commerce transactions, were produced by different software components, wherein a software component is interpreted to include an application*) that are foreign to the machine and the machine-readable medium (*paragraph [0004] first and second sentences; paragraph [0006], first sentence; paragraph [0008], "other interchangeable structured document formats". Since the system of Manning stores structured documents in multiple structured document formats available over the Internet, it would have been obvious that the formats were foreign to the system*) the software components having persistent data in different model structures (*paragraph [0004] first and second sentences; it would have been obvious that persistent data from vector graphics is in a different model structure than e-commerce transactions. ; paragraph [0008], "other interchangeable*

structured document formats" are different model structures), the persistent data relating to diverse types of objects (***paragraph [0004]***).

8.4.3. receiving metadata comprising at least in part a description of the model structure (***figure 3, items 100 - 102; and paragraph [0028]***), the metadata describing the persistent data (***paragraph [0004]***).

8.4.4. establish, using the metadata and without the using the software component from which the persistence package was received, during a runtime of the machine, a storage format and a storage location for the persistent data (***paragraph [0028]; paragraph [0041]; figure 3, elements 102 - 110; paragraph [0009], "At least one table is generated based on a schema of elements . . . "; please note that a schema determines a data format and a table location***).

8.4.5. apply the established storage format to the persistent data to format the persistent data for storage (***figure 3, element 124, since the accessed object is stored, it would have been obvious that the established storage format is applied; and paragraph [0029], since each object (e.g. attribute value or content) is stored in an element table, it would have been obvious that an established storage format is applied***) from the format of the software component into a storage format that is compatible with the machine and with a storage device independent of the software component (***paragraph [0022], paragraphs [0027] - [0029], and paragraph [0034]; it would have been obvious that the received data such as vector graphics and e-commerce transactions is being stored in a relational database, which is a storage format that is compatible with the receiving system and with a storage device independent of the software component***).

8.4.6. **Manning does not specifically teach**:

8.4.6.1. receive metadata comprising at least in part a description of the model structure, the metadata describing the persistent data ***and comprising, at least in part, a description of the format of the persistent data;***

8.4.7. **Maimone appears to teach**:

8.4.7.1. receive metadata comprising at least in part a description of the model structure, the metadata describing the persistent data ***and comprising, at least in part, a description of the format of the persistent data*** (***figure 1, element 18, and column 3, lines 60 - 67, and column 4, lines 1 - 16; it would have been obvious that attribute type of element 18 was a description of persistent data***).

8.4.8. The motivation to use the art of Maimone with the art of Manning would have been the benefit recited in Maimone that the invention improves the ability to persist objects of an object-oriented environment in a relational database (***column 2, lines 24 - 27***).

8.4.9. Obviousness must be determined in light of the knowledge of the ordinary artisan. The following reference teaches knowledge of the ordinary artisan at the time of invention:

8.4.9.1. Sams, "Sams Teach Yourself PL/SQL™ in 21 Days", second edition, 2000, Sams Publishing; teaches a database schema establishing a storage location (table) and data storage formats for the data in the table (page 224, listing 8.1 Creating the Employee Table), and teaches automatic data format conversion by a database system (*page 224 – 225, section "Using the INSERT Statement", please note that the pay rate being inserted into the employee table is a text ('8.50') but the pay rate is inserted as a number in the table, where the pay_rate is defined as a number in the schema on page 224, listing 8.1 Creating the Employee Table*).

8.4.9.2. Morgenstern (U.S. Patent No. 5,970,490) teaches using metadata to build a transform to convert data from one format into another format.

8.4.10. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Maimone and with the art of Manning to produce the claimed invention.

8.5. Regarding claim 44, Manning appears to teach instructions ***(paragraphs [0021] and [0022])***, that when executed, cause a machine to store the persistent data using the storage format ***(figure 3, item 124; and paragraph [0029])***.

8.6. Regarding claim 47, Manning appears to teach instructions, that when executed cause a machine to retrieve the persistent data using the storage format ***(figure 4, all items; and paragraph [0030])***.

8.7. Regarding claim 48, Manning appears to teach instructions, that when executed, cause a machine to select and/or create, based on the metadata, a transform to establish at least one of the storage format and the storage location ***(figure 3, items 102 - 110; and paragraph [0028], sentences 1 - 4)***.

8.8. Regarding claim 49, Manning appears to teach receiving persistent data compatible with one of any type of processor, any type of programming language, any type of operating system, and any type of architecture ***(paragraph [0021])***.

9. **Claims 45 and 46** are rejected under 35 U.S.C. 103(a) as being unpatentable over Manning as modified by Maimone as applied to claims **43 - 44 and 47 - 49** above, in view of XML ("Extensible Markup Language (XML) 1.0"; W3C Recommendation 10-Feb-98, 1998).

9.1. Manning as modified by Official Notice and Maimone teaches receiving persistent data having a model structure as recited in claims **43 - 44 and 47 - 49** above.

9.2. Claim 45 is a dependent claim of claim 43, and thereby inherits all of the rejected limitations of claim 43.

9.3. Claim 46 is a dependent claim of claim 45, and thereby inherits all of the rejected limitations of claim 45.

9.4. The art of Manning as modified by Maimone is directed toward a method, system, program, and data structures for managing structured XML documents in a database ***(Manning, Title and Abstract; and paragraph [0020] regarding the XML document)***.

9.5. The art of XML is directed toward describing the Extensible Markup Language (XML) ***(Abstract)***.

9.6. Regarding claim 45, Manning appears to teach receiving metadata ***(figure 3, item 100; and paragraph [0004] - please note that an XML document contains both persistent data and metadata)***.

9.7. Regarding claim 46, Manning appears to teach receiving a persistence package comprising persistent data and metadata ***(figure 3, item 100; and paragraph [0004] - please note that an XML document contains both persistent data and metadata)***, and to extract the persistent data and the metadata from the persistence package (***paragraphs [0028] and [0029]; and figure 3, all elements***).

9.8. Regarding claim 45, Manning as modified by Maimone does not specifically teach receiving metadata ***conforming to a metadata template comprising rules for describing a data model structure of the persistent data***.

9.9. Regarding claim 45, Manning appears to teach that the metadata received in claim 45 conforms to a metadata template comprising rules for describing a data model structure of the persistent data *(page 2, section 2. Documents, first sentence; and page 3, section 2.1 Well-Formed XML Documents)*.

9.9.1. Regarding *(page 2, section 2. Documents, first sentence; and page 3, section 2.1 Well-Formed XML Documents)*; the reference XML describes the rules that the metadata conforms to, and specifically the production in section 2.1 is a metadata template.

9.10. The art of XML and the art of Manning are analogous art because they both contain the art of interpreting XML documents.

9.11. The motivation to use the art of XML with the art of Manning as modified by Maimone would have been obvious given the need in Manning to interpret XML documents, and the rules given in XML to form valid XML documents.

9.12. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of XML with the art of Manning as modified by Maimone to produce the claimed inventions.

10. **Examiner's Note:** Examiner has cited particular columns and line numbers in the references applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant in preparing responses, to fully consider the references in their entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the Examiner.

Allowable Subject Matter

11. Claims 1 - 5, 7 - 42 appear to be allowable over the prior art of record.

12. Following is a statement of Examiner's reasons for indicating allowable subject matter.

12.1. Regarding claim 1, while Manning appears to teach:

12.1.1. receiving a persistence package at a running system from one of a plurality of different software the persistence package including persistent data and metadata, the software components having persistent data in different formats ~~that are foreign to the running system~~;

12.1.2. extracting persistent data and metadata from the persistence package, the persistent data relating to diverse types of objects constructed at runtime of the software component and needed during more than one invocation of the software component, the metadata describing the persistent data ~~and comprising, at least in part, a description of the format of the persistent data~~.

12.1.3. establishing, based on the extracted metadata, ~~a transform for~~ a storage format and a storage location for the persistent data;

12.1.4. applying *a schema* ~~the transform~~ to the persistent data to format the persistent data without using the software component from which the persistence package was received from the format of the software component into a storage format that is compatible with the receiving system and with a storage device of the running system independent of the software component;

12.1.5. storing the persistent data in the storage device in the storage format,

12.2. and SqlDatabaseLanguage appears to teach:

12.2.1. ***a transform for*** a storage format for persistent data (***page 34, section 4.8.5 Transforms for user-defined types, "This SQL-invoked function maps the user-defined type value into a value of an SQL pre-defined type, and gets invoked whenever a user-defined type value is passed to a host language program or an external routine"***).

12.2.2. applying ***the transform*** (***page 34, section 4.8.5 Transforms for user-defined types, "This SQL-invoked function maps the user-defined type value into a value of an SQL pre-defined type, and gets invoked whenever a user-defined type value is passed to a host language program or an external routine"***),

12.3. and Morgenstern appears to teach:

12.3.1. ***a transform for*** a storage format for persistent data (***figure 2, at least elements 42, 46, 22, 32, 36, 23, 66, and column 8, lines 53 - 67, column 9, lines 1 - 3, and column 7, lines 16 - 67, and column 8, lines 1 - 53***),

12.4. and Maimone appears to teach:

12.4.1. the metadata describing the persistent data, ***and comprising, at least in part, a description of the format of the persistent data*** (***figure 1, element 18, and column 3, lines 60 - 67, and column 4, lines 1 - 16; it would have been obvious that attribute type of element 18 was a description of persistent data. Please note that Manning also shows metadata that comprises a description of the format of the persistent data in figure 5, for example, element 1004 describing the string as a TextObject, and showing the arrangement of fields (a format).***),

12.5. none of these references, either alone or in combination, appear to teach a method specifically including:

12.5.1. Regarding claim 1, "software components ~~having persistent data in different formats~~ that are foreign to the running system", in combination with the remaining feature and elements of the claimed invention.

12.6. Regarding claim 9, while Manning appears to teach:

12.6.1. a data storage device, a running receiving system coupled to the data storage device, the receiving system including a persistence engine to receive a persistence package from one of a plurality of different software components ~~that are foreign to the running system~~, the persistence package including persistent data and metadata, ~~the metadata comprising, at least in part, a description of the format of the persistent data~~ the software components having persistent data in different formats, wherein the persistence engine extracts persistent data and metadata from the persistence package, wherein the persistence engine uses the extracted metadata passed from the persistence package to establish, without using the software component from which the persistence package was received, a storage format and location to store the persistent data in the data storage device, and wherein the persistence engine applies the storage format to the persistent data to format the persistent data from the format of the software component into a storage format that is compatible with the receiving system and with the storage device independent of the software component, and to store the formatted system data in the data storage device,

12.7. and Maimone appears to teach:

12.7.1. the metadata comprising, at least in part, a description of the format of the persistent data,

12.8. none of these references, either alone or in combination, appear to teach an apparatus specifically including:

12.9. Regarding claim 9, "software components that are foreign to the running system", in combination with the remaining feature and elements of the claimed invention.

12.10. Regarding claim 29, while Manning appears to teach:

12.10.1. a communications interface; a data model description receiver to receive a data model description from one of a plurality of different software components ~~that are foreign to the apparatus~~, the software components having persistent data in accordance with different data models, the data model descriptions describing, at least in part a format of data associated with the models; a set of transforms; a transform generator having an assembler to produce a transform based on the data model description ~~and the comparison~~ independent of the software component from which the data model description was received, the transform establishing a storage format and a storage location for data associated with the model; a storage device; a transform engine to apply a transform to format persistent data for storage from the format of the software component into a storage format that is compatible with the storage device independent of the software component; and a storing interface to store the formatted persistent data in the storage device,

12.11. and Maimone appears to teach:

12.11.1. a data model comparator to produce a comparison independent of the software component from which the data model description is received between the data model description and a data model in a transform in the set of transforms; a transform generator having an assembler to produce a transform based on the data model description and the comparison,

12.12. none of these references, either alone or in combination, appear to teach an apparatus specifically including:

12.13. Regarding claim 29, "software components that are foreign to the apparatus", in combination with the remaining feature and elements of the claimed invention.

12.14. Regarding claim 29, while Manning appears to teach:

12.14.1. receiving a data model description at a running system from one of a plurality of different software components ~~that are foreign to the running system~~, the software components having persistent data in accordance with different data models, the persistent data relating to diverse types of objects, ~~the data model description describing the persistent data, and comprising, at least in part, a description of the format of the persistent data~~;

12.14.2. comparing the data model description to a preexisting data model independent of the software component from which the data model description is received;

12.14.3. assembling a transform at the running system independent of the software component from which the data model description is received based on the data model description ~~and the comparison~~ to establish a storage format for persistent data during runtime of a system;

12.14.4. applying a transform to format persistent data for storage from the format of the software component into a storage format that is compatible with a storage device independent of the software component; and storing the formatted persistent data at the data storage device.

12.15. and Morgenstern appears to teach:

12.15.1. assembling a transform based on a comparison,

12.16. and Maimone appears to teach:

12.16.1. software components that are foreign to the running system,

12.16.2. the data model description describing the persistent data, and comprising, at least in part, a description of the format of the persistent data;

12.17. none of these references, either alone or in combination, appear to teach a method specifically including:

12.18. Regarding claim 36, "software components that are foreign to the running system", in combination with the remaining feature and elements of the claimed invention.

Conclusion

13. The prior art made of record in a prior Office action, and not relied upon, is considered relevant to the Applicant's disclosure:

13.1. Barbara Staudt Lerner et al.; "Beyond Schema Evolution to Database Reorganization", 1990, Proceedings of the European conference on object-oriented programming on Object-oriented programming systems, languages, and applications OOPSLA/ECOOP '90, volume 25, issue 10, pages 67 - 76; teaches transforming classes in a database to affect the contents as little as possible (page 70, section 2.4).

13.2. L.M. Haas et al.; "Transforming Heterogeneous Data with Database Middleware: Beyond Integration", 1997, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, pages 1 - 6; teaches schema transformation.

13.3. Elke A. Rundensteiner et al; "Maintaining Data Warehouse over changing information sources", June 2000, Communications of the ACM, Volume 43, Number 6, pages 57 - 62; teaches evolving schemas in data warehouses.

14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Russ Guill whose telephone number is 571-272-7955. The examiner can normally be reached on Monday - Friday 10:00 AM - 6:30 PM.

15. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Paul Rodriguez can be reached on 571-272-3753. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Any inquiry of a general nature or relating to the status of this application should be directed to the TC2100 Group Receptionist: 571-272-2100.

16. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Russ Guill

Examiner

Art Unit 2123

RG

PAUL RODRIGUEZ
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100